

FLIGHT EXPERIMENT ON A FLIGHT PATH OPTIMIZATION ALGORITHM FOR AIRCRAFT IN TROUBLE

Kazuya MASUI*, Hiroshi TOMITA*, Yutaka KOMATSU**

*Japan Aerospace Exploration Agency (JAXA), **the University of Tokyo

Keywords: flight test, simulation, guidance, flight path, optimization

Abstract

The Japan Aerospace Exploration Agency (JAXA) and the University of Tokyo are jointly researching an on-line four-dimensional flight path optimization system for aircraft in trouble.

Taking into account flight constraints that depend on aircraft failure conditions, the proposed algorithm successively generates optimal flight path and airspeed commands until the aircraft is guided to a runway. By combining the RTA algorithm and the R-TABU search method, this algorithm can find the optimal four-dimensional flight path in real time, responding to changes in the failure conditions.*

The applicability of the generated flight path to real flight was evaluated as the first of a series of flight experiments of this algorithm. Responding to changes in failure or flight conditions, a reasonable flight path was generated that could be tracked by pilots using a tunnel-in-the-sky flight guidance display. Several technical issues revealed through the flight experiments are also discussed.

1 Introduction

The amount of air transportation continues to increase steadily, while the fatal accident rate has remained almost constant for the past twenty years. Further reducing the fatal accident rate is therefore recognized to be essential for future air transportation. Aiming to prevent in-flight failures from leading to fatal accidents, the Society of Japanese Aerospace Companies (SJAC) is promoting research on an autonomous flight control and guidance system for aircraft in trouble [1]. This system consists

of three components; the first component detects the occurrence of a failure, the second component identifies the equipment or system affected and reconfigures the flight control system to stabilize the aircraft and to compensate for the failure, and the third component generates a four-dimensional flight path for emergency landing, which is optimized according to the failure conditions, and guides the aircraft to a runway. This paper describes flight experiments to evaluate the applicability of an on-line four-dimensional flight path optimization algorithm which has been newly developed as a part of the autonomous flight control and guidance system for aircraft in trouble.

The on-line flight path optimization algorithm is required to calculate an optimal flight path for prescribed criteria in real time and to be robust against various flight conditions. By combining the RTA* (RTA star) algorithm [2] and the R-TABU search method [3], the University of Tokyo has developed an on-line four-dimensional flight path optimization algorithm which can account for changes in aerodynamic characteristics and the remaining control capability of an aircraft in trouble. To evaluate the applicability of this new algorithm to real flight situations, the Japan Aerospace Exploration Agency (JAXA) has performed a series of flight experiments with its in-flight simulator 'MuPAL- α ' [4]. As the first stage of this flight evaluation, two test pilots tracked flight path and airspeed commands generated by the algorithm, which were displayed as flight guidance information using a tunnel-in-the-sky image [5]. Through landing approaches under various conditions to a

‘virtual runway’ in the air, instructive pilot comments and data were obtained.

2 Flight Path Optimization Algorithm

The generated flight path is expressed as a series of nodes linked by segments that trace a trajectory in space between the nodes. A node on the flight path is a four-dimensional vector of horizontal position, altitude, and time. The initial state is the aircraft’s current position and time, and the terminal node is at the destination runway threshold.

The flight path is updated at a prescribed interval by searching for an optimal ‘sub-node’ at some time ahead between the current (initial) node and the terminal node, and determining the optimal trajectory to reach that sub-node.

2.1 Optimization Algorithms

2.1.1 The RTA* Algorithm

The RTA* algorithm is an on-line algorithm that solves a state-space search problem. This problem finds a path to minimize the cost in going from an initial state to a terminal state by expanding nodes iteratively in the ‘state set’, which is a discrete set of probable states including the initial and terminal states.

The RTA* algorithm performs the search processes efficiently by using a ‘heuristic’, a pre-experiential and intuitive rule based on experience, as shown in Fig. 1. At each iteration in the search for an optimal path from the

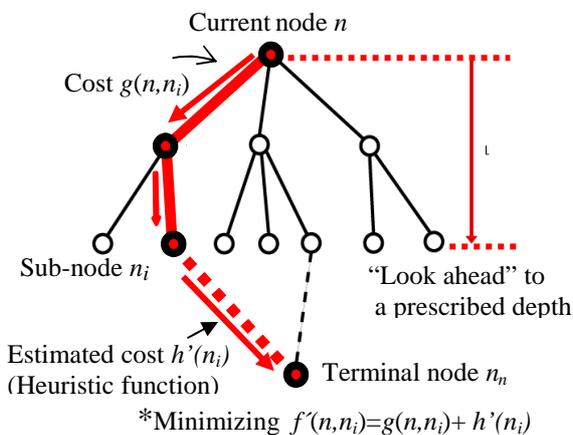


Fig. 1. RTA* Algorithm

current node n to the terminal node n_n , the RTA* algorithm ‘looks ahead’ along paths to a sub-node n_i at a prescribed depth and places the nodes on a path to minimize $f'(n, n_i) = g(n, n_i) + h'(n_i)$, where $g(n, n_i)$ is the cost from the current node n to the sub-node n_i and $h'(n_i)$ is the heuristic function, which is the estimated cost of going from sub-node n_i to the terminal node n_n . The result and the efficiency of the heuristic search depend on the design of heuristic function $h'(n_i)$.

2.1.2 The R-TABU Search Method

The R-TABU search method is an iterative method for solving non-linear programming problems. This divides the search area into multiple areas of different size and searches for the optimal solution randomly, which results in the random generation of trial solutions around an initial solution. The optimum of these trial solutions is then selected as the initial solution for the next iteration. As iterations progress, the optima are found in narrower areas and the cost function is reduced (Fig. 2).

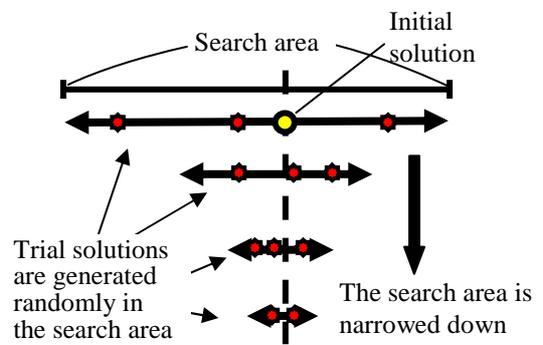


Fig. 2. R-TABU Search Method

While on the one hand an intensive search in the vicinity of the initial solution would prevent a blind search and realize stable convergence, this method can find the globally optimal solution because it tries to search other areas if the cost function is not reduced within a specified number of trials. As the trial solutions are generated from random numbers, the R-TABU search method can reach the optimal solution more quickly than methods that use the gradient of the cost function.

2.2 Aircraft Dynamics Model

In the flight path optimization problem, the aircraft is considered as a point mass subjected to aerodynamic forces, gravity and thrust [6]. Engine thrust, angle of attack, sideslip angle and bank angle are used as control inputs. The components of aerodynamic force, lift L , drag D and side force Y , are modeled as

$$L = \frac{1}{2} \rho V^2 S \cdot C_L(\alpha, C_T) \quad (1)$$

$$D = \frac{1}{2} \rho V^2 S \cdot C_D(\alpha) \quad (2)$$

$$Y = \frac{1}{2} \rho V^2 S \cdot C_Y(\alpha, \beta, C_T) \quad (3)$$

where C_T is the thrust coefficient. The aircraft mass m is related to thrust T as

$$\frac{d}{dt} m = -\mu \cdot T \quad (4)$$

where μ is the rate of fuel consumption.

2.3 Flight Path Structure

A series of sinusoidal functions are used as flight path elements interpolating the nodes:

$$x(t) = A_0 + \sum_{k=1}^N (A_k \cos k\omega t + B_k \sin k\omega t) \quad (5)$$

$$y(t) = C_0 + \sum_{k=1}^N (C_k \cos k\omega t + D_k \sin k\omega t) \quad (6)$$

$$z(t) = E_0 + \sum_{k=1}^N (E_k \cos k\omega t + F_k \sin k\omega t) \quad (7)$$

The use of sinusoidal functions allows higher order differentiation. States variables such as speed, acceleration, angle of attack, sideslip angle, attitude angles and thrust are calculated directly from the generated flight path. By matching the flight path segments up to the third order of differentiation at each node, the transient at each waypoint (node) is suppressed.

2.4 Flight Path Generation

At each current node n , according to the current position and velocity, the flight path that minimizes the cost function $f'(n, n_i)$ is determined by optimizing the position and velocity at subnode n_i and the series of sinusoidal functions which interpolate the subnodes. As the subnode n_i is set at a prescribed depth, the adjacent waypoint, which will be considered as the new current node in

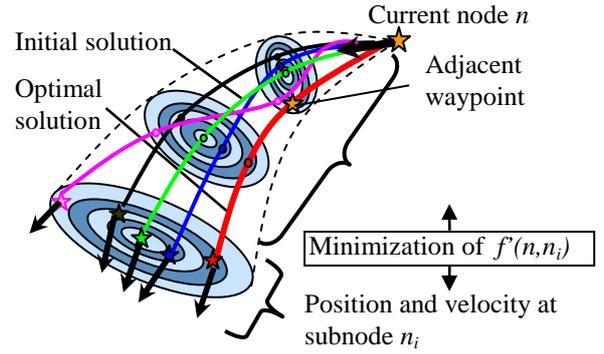


Fig. 3. Flight Path Generation

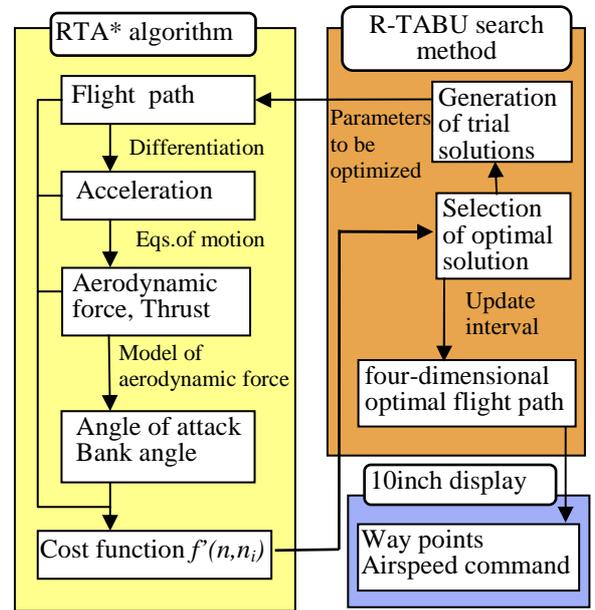


Fig. 4. Flight Path Optimization Algorithm

the next iteration, is located on the flight path interpolating the sub-nodes (Fig. 3). The R-TABU search method generates trial solutions as a discrete set of probable states and selects the optimal flight path, while the RTA* algorithm calculates the cost function as shown in Fig. 4.

3 Flight Experiment Method

3.1 MuPAL- α and its Experiment Systems

JAXA has been operating its in-flight simulator 'MuPAL- α ' (Fig. 5), which is based on a Dornier 228-202, since April 2000. 'MuPAL' is the abbreviation of 'Multi-Purpose



Fig. 5. MuPAL- α

Aviation Laboratory’ and ‘ α ’ is the first letter of the Greek word representing an airplane. In addition to the original mechanical flight control system, MuPAL- α is equipped with an experimental fly-by-wire (FBW) flight control system that includes a Direct Lift Control (DLC) system. These systems give MuPAL- α a variable stability and response capability, which is applicable not only to motion simulation but also to the flight demonstration of advanced guidance and control technologies. Several types of visual information system are prepared for the pilot interface, and a high accuracy data acquisition system is also installed.

MuPAL- α is operated by two pilots; a safety pilot operates the aircraft using the original mechanical flight control system, and an evaluation pilot flies with the FBW flight control system. As a feature of MuPAL- α , a researcher can freely design the guidance and control laws for the FBW computer and the display formats for the evaluation pilot without any impact on airworthiness. Further, the FBW computer can exchange data and commands with personal computers (PC) which may be



Fig. 7. Tunnel-In-the-Sky Image

programmed by researchers. This flexibility allows rapid development and modification of control laws and display formats according to the results of flight experiment.

Fig. 6 shows an outline of the experiment systems used in this research. The FBW computer generates commands for electric actuators based on the evaluation pilot’s control inputs. In the first series of flight experiments, the aerodynamic characteristics and the control derivatives were assumed to be identical to the original Do228, and so the DLC system was not activated. For the next series of flight experiments, the FBW computer will be programmed to simulate changes of aerodynamic characteristics and control derivatives caused by failures. The on-line four-dimensional flight path optimization algorithm is executed on a separate guidance computer (Guidance PC) and generates the optimal waypoints and airspeed commands. A further PC generates the visual display indicating flight parameters and guidance for the evaluation pilot, which is displayed in the cockpit on a 10-inch

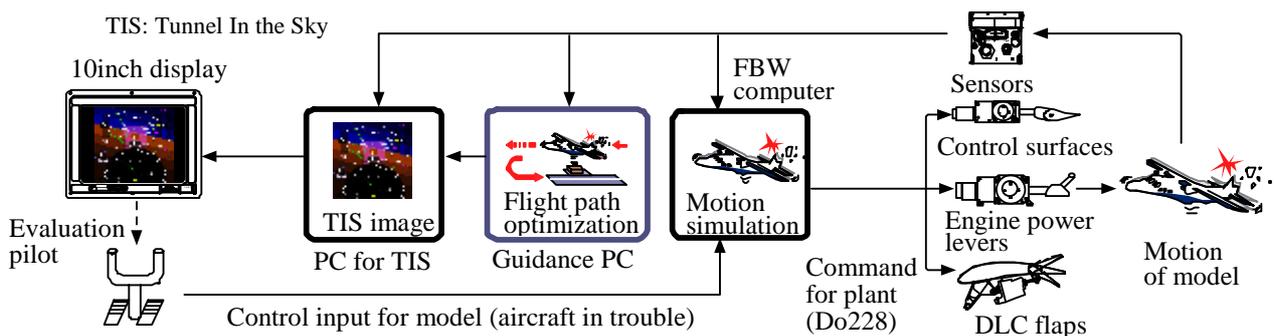


Fig. 6. Outline of the Experiment Systems for the On-Line Flight Path Optimization Algorithm

display. The optimal waypoints are presented by using a tunnel-in-the-sky image (Fig. 7), while the airspeed command is presented as a numeric readout and a pentagonal symbol on the airspeed tape. When the commanded airspeed increases or decreases, a triangular symbol blinks above or below the numeric readout. The evaluation pilot manually controls the aircraft to track the center of the guidance tunnel and the commanded airspeed.

3.2 Flight Experiment Method

In the first series of flight experiments, the evaluation pilot manually tracked the optimal four-dimensional flight path to a virtual runway in the air. The runway was not fixed above a prescribed point on the earth, but its location and azimuth were automatically set according to the aircraft’s position and its heading at the instant the guidance function was started. This method maximized the number of cases that could be evaluated within a limited period of flight experiment time.

The initial and terminal conditions were set as shown in Table 1. To confirm that the proposed algorithm could generate an optimal flight path even if a complicated shape were required, the azimuth of runway was set so that the aircraft had to change its initial heading

Table 1. Initial and Terminal Conditions

| Parameter | Initial cond. | Terminal cond. |
|--------------|-----------------|-----------------|
| Position x | 15000 m | 0 m |
| y | 10000 m | 0 m |
| z | 1000 m | 100m |
| Airspeed | 120 kt (62 m/s) | 114 kt (59 m/s) |
| Heading | 170 degrees | 90 degrees |
| Path angle | 0 degree | -1 degree |

Table 2. Constraint Conditions

| Parameter | Lower limit | Upper limit |
|-----------------|-----------------------------|--------------|
| Airspeed | 90kt(46m/s) | 140kt(72m/s) |
| Angle of attack | -5 degrees | 15 degrees |
| Pitch angle | -10 degrees | 20 degrees |
| Bank angle | -20 degrees or -7degrees | 20 degree |
| Load factor | 0.5 G | 1.5G |
| Altitude | 0 m | not limited |

more than 270 degrees to reach it.

The cost function to be minimized consisted of time and fuel consumption to reach the runway. The constraint conditions were set as shown in Table 2. The thrust constraints were approximated by a linear function of airspeed. These constraints were added to the cost function by the penalty function method. As an example failure case, the aircraft’s left turn capability was constrained by limiting left bank angle to no greater than 7 degrees. The actual initial altitude was set at between 6500 ft (1980 m) and 8500 ft (2590m).

3.3 Pre-Flight Simulation

At first, it was confirmed by computer simulation that the proposed algorithm could steadily generate an optimal flight path within a specified time (nominally 30 s) and that the generated guidance would result in the aircraft staying within all the imposed constraint conditions throughout the flight.

Before each flight experiment, pilot-in-the-loop simulations were carried out using JAXA’s ‘Flight Simulator Complex for Advanced Technology’ (FSCAT), which was established in August 2003 as a new simulator facility for research. For the simulation of the proposed algorithm, a generic fixed-wing aircraft cockpit was used, without its hydraulic cockpit motion system activated. The Guidance PC was connected to the FSCAT flight dynamics computer and generated the four-dimensional optimal flight path in real time. Flight parameters and guidance were depicted on a display in the simulator cockpit with the same format as in the flight experiment. In these simulations, three JAXA test pilots evaluated the following items:

- Format of the optimal flight path and airspeed command displays
- Update rate of the optimal flight path and airspeed command
- Robustness against various flight conditions
- Appropriateness of the generated flight path and airspeed commands for manual flight control

- Tracking errors against the optimal flight path and airspeed commands
- Deviation from each constraint condition
- Pilot workload to track the optimal flight path and airspeed commands
- The effects of steady wind

Flight evaluation of the algorithm was judged to be practicable after some minor changes based on the pilot-in-the-loop flight simulation results. However the pilots commented that the thrust control workload to track the airspeed commands was acceptable but considerably high compared with normal operations. Therefore, evaluation of additional constraints to reduce thrust control was added to the flight experiment objectives.

4 Flight Experiment Results

4.1 Outline of Flight Experiments

4.1.1 Flight Experiment Cases

Six flights were performed to evaluate the on-line four-dimensional flight path optimization algorithm from October 2003 to February 2004. Two JAXA test pilots participated as evaluation pilots. Six or seven cases were evaluated in each flight of about two hours.

During the first three flights, the following were evaluated.

- Formats of the optimal flight path and airspeed command displays
- Effect of different flight path update rates
- Robustness against actual flight conditions
- Deviation from each constraint condition
- Preliminary evaluation of tracking error, pilot workload and effects due to steady wind

After some minor changes mentioned in sections 4.2.1 and 4.2.2 below, a further three flights were performed to evaluate tracking errors against the optimal flight path and airspeed commands, pilot workload and deviation from each constraint condition, if any. The following were considered as parametrical conditions.

- Left bank angle limit due to a supposed aircraft failure
- Thrust control constraints
- Direction of steady wind at the initial position
- Considerations for steady wind

To evaluate the last condition, the proposed algorithm was modified before the last flight.

Table 3 shows the combinations of parametrical conditions evaluated in the last three flights. Here, the wind direction at the initial point was used to specify the direction of steady wind expected on the longest decent leg, which was flown at a heading opposite to the runway azimuth. In case of a wind from the right at the initial point, a headwind would be expected on the longest decent leg if the wind direction remained constant.

4.1.2 Typical Flight Test Data

Fig. 8 shows a typical flight path recorded for a case without the left bank constraint (Case A1 in Table 3). The origin of the inertial reference frame is located at the touchdown point on the virtual runway with the x -axis along the runway centerline. Fig. 8 also shows the time histories of representative variables measured in the same case, including indicated airspeed (IAS , dotted line: command, solid line: actual value), airspeed deviation from the command value (ΔIAS), deviation from the optimal flight path ($\Delta Path$), angle of attack (α), pitch angle (θ), bank angle (ϕ), vertical load

Table 3. Combinations of Parametrical Conditions

| Case No. | A1, A2 | B1 | C1, C2 | D1- D4 | E1 | F1, F2 | G1, G2 | H1 | J1 | K1 |
|-------------------------------------|-------------|------|-----------|-----------|------|-----------|-------------|-------|---------|-------|
| Wind direction at the initial point | right | left | right | right | left | right | left | right | left | right |
| Left bank angle limit (degrees) | -20 | | -7 | -20 | | -7 | -20 | | | |
| Constraints for thrust control | not applied | | | applied | | | not applied | | applied | |
| Considerations for steady wind | not applied | | | | | | applied | | | |

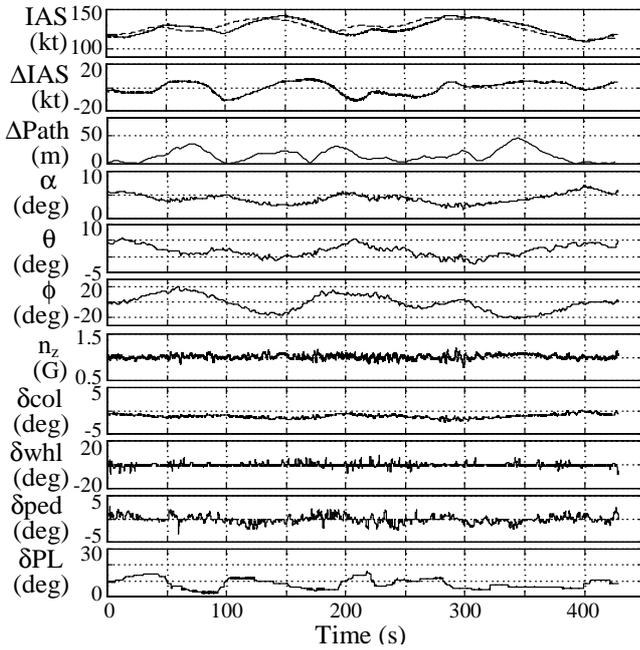
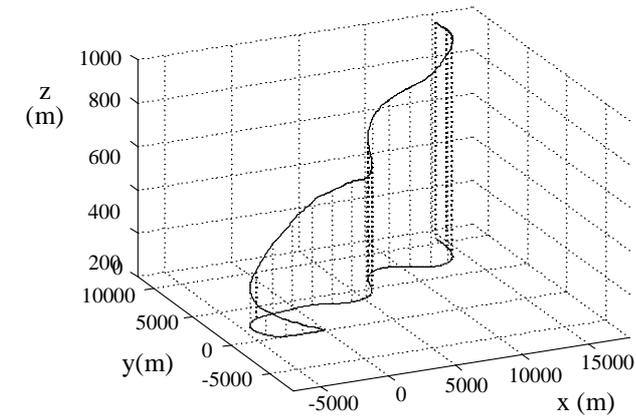


Fig. 8. Flight Path and Time Histories (CaseA1)

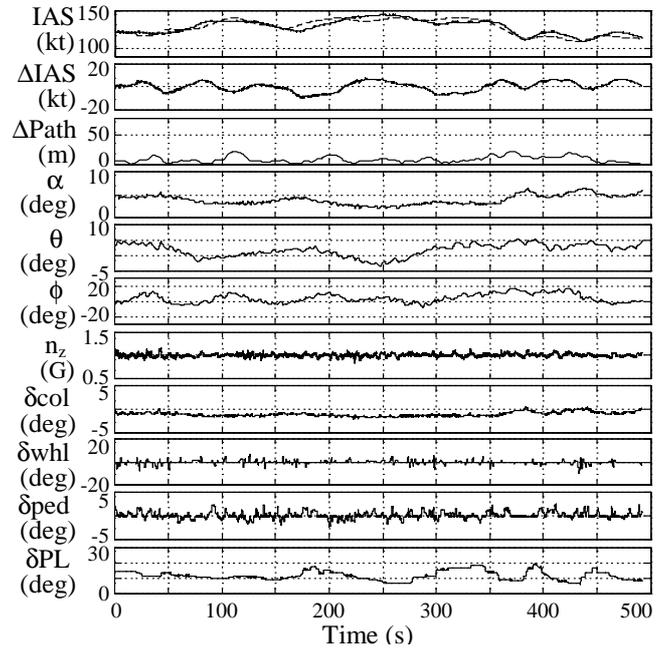
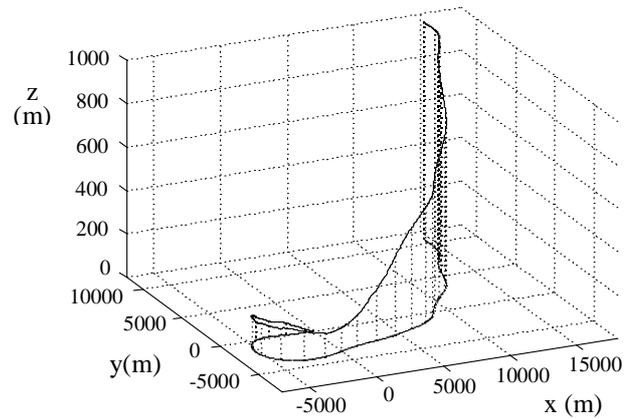


Fig. 9. Flight Path and Time Histories (CaseC2)

factor (n_z) and the evaluation pilot's control inputs: column angle (δcol), wheel angle (δwhl), rudder pedal angle (δped) and engine power lever angle (δPL). Here, $\Delta Path$ means the distance between the current aircraft position and the nearest point on the whole optimal flight path. Fig. 9 shows a typical flight path and time histories recorded for a case with restricted left bank angle (Case C2 in Table 3).

4.2 Technical Considerations

4.2.1 Airspeed command display format

At first, the airspeed to achieve 3 s from the present time was indicated as the target airspeed command. However, during the first flight the pilot commented that the delay of

thrust control made tracking the target airspeed difficult because the direction of change of the target airspeed (increase or decrease) could not be predicted from the target value itself.

To address this, a symbol to indicate the change direction of target speed (increase: upward triangle, decrease: downward triangle) was added (Fig. 7), and the pilots commented that this reduced workload to track the target airspeed. A reduction in airspeed deviation was also confirmed from the measured data.

4.2.2 Optimal flight path update rate

If the generated flight path is updated at intervals that are too short to allow the optimization algorithms to perform a sufficient number of iterations, only an interim solution may be presented. On the other hand, if the

update interval is too long, there may be significant changes to the generated flight path on each update and the updates will not be smooth.

Different update rates were tried in the flight experiments, and it was confirmed that the nominal update rate (30 s) could generate reasonable and smooth flight paths.

4.2.3 Robustness against actual flight conditions

In all flight experiment cases, the proposed algorithm never failed to reach the runway, and no discontinuity or unsmooth change of generated flight path was found. The proposed algorithm has been proved to be robust against the actual flight conditions.

4.2.4 Appropriateness of generated flight path and airspeed commands

The flight experiments evaluated whether the commanded flight path could be tracked manually while remaining within the constraint limits, which represented the assumed level of damage.

In almost all cases, there was no deviation from any of the constraint conditions in Table 2. For the case of a normal, undamaged aircraft, bank angles near the 20-degree constraint limit were used in both turn directions as shown in Fig. 8. On the other hand, for the case assuming a damaged aircraft, the left bank angle was smaller than the 7 degree limit as shown in Fig. 9, while the time to reach the runway was 65 seconds longer compared with Fig. 8. Further, as mentioned in section 4.2.5 below, tracking errors against the optimal flight path and airspeed commands were within an acceptable range for all cases. It was thus demonstrated that the proposed algorithm could generate appropriate flight path and airspeed commands which satisfied the prescribed constraints.

However, it was found that the bank angle constraint (20 degrees) was exceeded momentarily in some cases, especially during the final turn to align with the runway. The steady wind and its changes were supposed to be one of the factors which might lead to bank angle constraints being exceeded. Section 4.2.7 below shows the flight experiment results of a

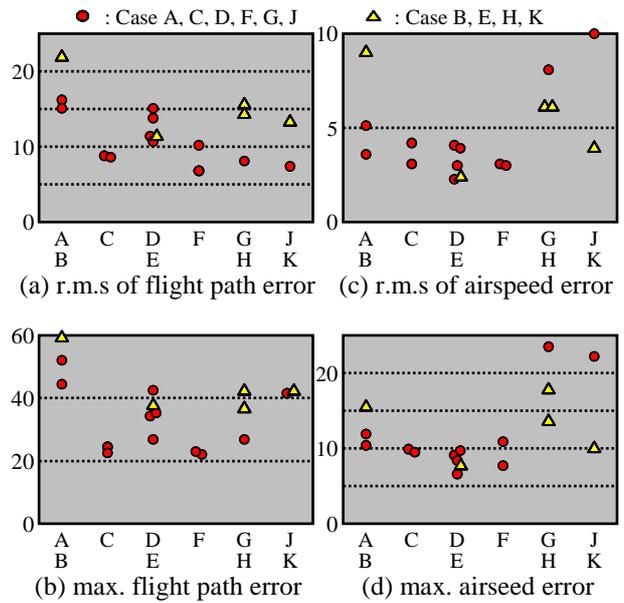


Fig. 10. Tracking Errors

modified algorithm that takes into account changes in the steady wind.

4.2.5 Tracking Errors

The r.m.s. and maximum values of tracking errors against the optimal flight path and airspeed commands are shown in Fig. 10. The r.m.s. flight path deviation is less than 22 m, and the maximum deviation is less than 60 m. The r.m.s. airspeed deviation is less than 10 kt (5.1 m/s). Excluding severe turbulence cases (Cases G–K), the r.m.s. airspeed deviation is almost less than 5 kt (2.6 m/s). Although the maximum value of airspeed deviation depends on the gust encountered in each case, it is less than 12 kt (6.2 m/s) in 70% of all cases.

The tracking errors in cases with a restricted left bank angle are less than those in other cases. It is supposed that the smoother flight path resulting from the left bank angle limit made manual tracking easier.

4.2.6 Thrust control constraints

The evaluation pilots commented that the thrust control workload was high because the target speed and sink rate changed so frequently that thrust had to be controlled much more carefully than in normal operations. Particularly in a situation where the vertical wind component fluctuated greatly over a small area,

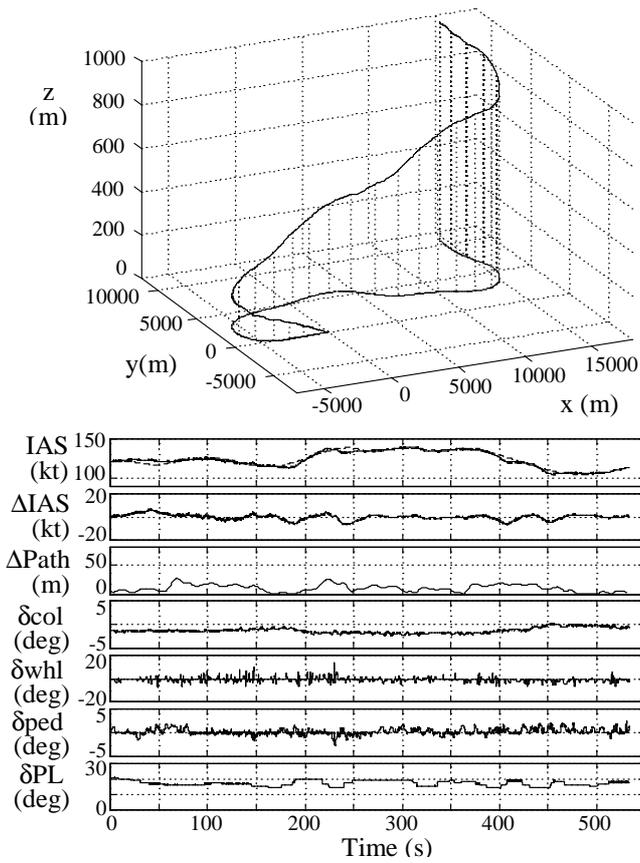


Fig. 11. Constraints for Thrust Control (CaseD3)

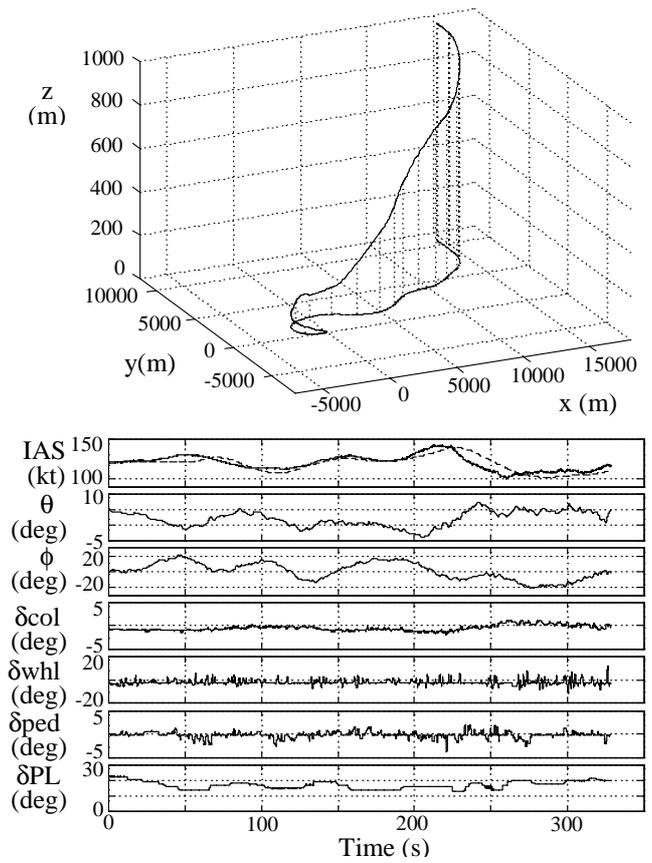


Fig. 12. Considerations for Steady Wind (CaseG2)

it was very difficult to track the indicated flight path precisely.

To reduce the amount of thrust control required, constraints on the range and rate of thrust change were added. Fig. 11 shows a typical flight path and time histories for cases with these constraints added. The additional constraints reduced the r.m.s. power lever stroke by about 18%, and the pilots commented that the thrust control workload was so much reduced that the pilot could give ample attention to other tasks. Tracking errors also decreased as shown in Fig. 10. Against this reduction of pilot workload, however, more time was needed to reach the runway.

4.2.7 Considerations for steady wind

To apply the on-line flight path optimization algorithm to actual aircraft operations, it is considered essential that the algorithm takes into account the steady wind condition and its changes. A headwind or tailwind may make the trim condition required to maintain airspeed and sink rate exceed the

capability of the aircraft, while a side wind may make the bank angle required to track a specified curve exceed the prescribed constraint as mentioned in the section 4.2.4.

To allow for the speed and direction of the steady wind, the proposed algorithm was modified so that the optimal flight path was calculated in a coordinate system fixed to the air and moving with the steady wind. The estimated values of three-dimensional components of steady wind obtained from MuPAL-α's air data and inertial sensors were averaged over 30 s. The optimal flight path was presented to the pilot after the transformation to the ground-fixed reference frame.

Typical flight path and time histories for cases taking steady wind into account are shown in Fig. 12. As shown in Figs. 13 and 14, conditions during the last flight flown in February were much more windy (average wind speed was 20.2 kt (10.4 m/s)) and gusty than the flights the previous autumn. Therefore, it is

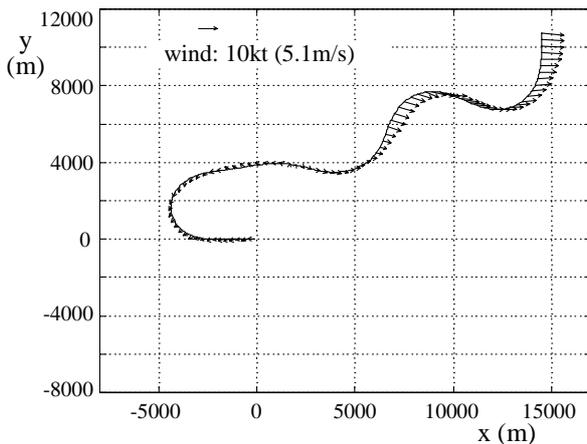


Fig.13. Wind Speed and Direction (Case A1)

difficult to directly compare the results of these flights.

In the case shown in Fig. 12, by considering the strong tailwind on the longest decent leg, the modified algorithm succeeded in reducing the time to reach the runway. The same effect was found also for cases with additional constraints to reduce thrust control workload. As for the tracking errors shown in Fig. 10, the differences due to the consideration of steady wind seem to be canceled by the effects of severe turbulence. The differences due to the reduction of thrust control workload were also not great, because the consideration of a strong tailwind reduced the thrust control required. However the effectiveness of taking steady wind into account was proved by the fact that the pilots commented as follows in spite of the windy conditions:

- Excessive bank angles were unnecessary.
- The indicated flight paths seemed more reasonable and fitted with the pilots' experiences.
- Control to track the indicated flight path became easier.

5 Conclusions and Future Plans

The applicability and robustness of the proposed on-line four-dimensional flight path optimization algorithm to real flight was demonstrated through the flight experiments, in which JAXA's pilots could track the generated

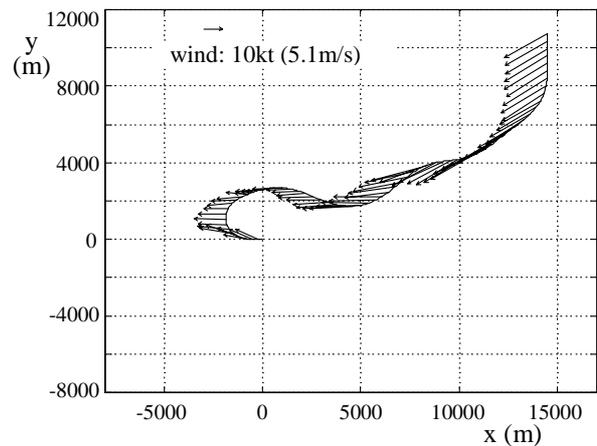


Fig.14. Wind Speed and Direction (Case G2)

flight path and airspeed commands with acceptable workload. It was confirmed that adding thrust control constraints might reduce pilot workload, and taking the steady wind conditions into account was effective in generating more practical flight paths.

As the next stage, flight experiments assuming various types of accidents or obstacles and the demonstration of an automatic flight control system to track the optimal flight path will be planned.

References

- [1] Suzuki S, Kawamura F and Masui K. Autonomous flight control and guidance system of accident aircraft. *Proc. ICAS2004*, Yokohama, 2004.
- [2] Korf R E. Real-time heuristic search. *Artificial Intelligence*, Vol. 42, No.2-3, pp 189-211, 1990.
- [3] Hu N. Tabu search method with random moves for globally optimal design, *International Journal for Numerical Methods in Engineering*, Vol. 35, pp 1055-1070, 1992.
- [4] Masui K and Tsukano Y. Development of a new in-flight simulator MuPAL- α . *Proc AIAA Modeling and Simulation Technologies Conference and Exhibit*, Denver, AIAA-2000-4574, 2000
- [5] Funabiki K, Muraoka K, Terui Y, Harigae M and Ono T. In-flight evaluation of tunnel-in-the-sky display and curved approach pattern. *Proc AIAA Guidance, Navigation and Control Conference and Exhibit*, Portland, AIAA-99-3966, 1999
- [6] Etkin B. *Dynamics of Flight*. John-Wiley and Sons Inc., 1959.